

ModuleRefGetOrResolveOpts

August 17, 2026

ModuleRefGetOrResolveOpts

Kind: Interface

Source: `packages/core/injector/module-ref.ts`

Part of: [Core](#)

`ModuleRefGetOrResolveOpts` configures how a provider is retrieved or resolved from a module reference. Use `strict` to control whether lookup is limited to the current module context, and `each` to request all matching provider instances instead of a single result.

Properties

Property	Type
<code>strict</code>	<code>boolean</code>
<code>each</code>	<code>boolean</code>

Diagram

```
graph LR
  A[ModuleRef.get / resolve] --> B[ModuleRefGetOrResolveOpts]
  B --> C{strict}
  C -->|true| D[Search current module context]
  C -->|false| E[Search broader module graph]
  B --> F{each}
  F -->|true| G[Return all matching instances]
  F -->|false| H[Return one matching instance]
```

Usage

```
import { ModuleRef } from '@nestjs/core';
```

```

class UserService {
  constructor(private readonly moduleRef: ModuleRef) {}

  getLocalRepository() {
    return this.moduleRef.get(UserRepository, {
      strict: true,
      each: false,
    });
  }

  getAllHandlers() {
    return this.moduleRef.get(APP_HANDLER, {
      strict: false,
      each: true,
    });
  }
}

```

AI Coding Instructions

- Pass `strict: true` when the dependency must be registered in the current module and should not be resolved from imported or global modules.
- Use `strict: false` for cross-module lookups, especially when resolving shared or globally available providers.
- Set `each: true` only for multi-provider tokens where multiple matching instances are expected.
- Handle the return shape carefully: `each: false` returns a single instance, while `each: true` returns a collection of instances.
- Keep lookup options aligned with the `ModuleRef.get()` or `ModuleRef.resolve()` call so provider scope and module visibility behave as intended.

Relationships

- IMPORTS → `IntrospectionResult`
- IMPORTS → `Scope`
- IMPORTS → `Type`